Journal of Korea Multimedia Society Vol. 19, No. 8, August 2016(pp. 1636-1645)
http://dx.doi.org/10.9717/kmms.2016.19.8.1636

수집한 GPS데이터의 상호방향성을 이용한 경로데이터 조합방법

구광민†, 박희민††

A Combination Method of Trajectory Data using Correlated Direction of Collected GPS Data

Kwang Min Koo†, Heemin Park††

ABSTRACT

In navigation systems that use collected trajectory for routing, the number and diversity of trajectory data are crucial despite the infeasible limitation which is that all routes should be collected in person. This paper suggests an algorithm combining trajectories only by collected GPS data and generating new routes for solving this problem. Using distance between two trajectories, the algorithm estimates road intersection, in which it also predicts the correlated direction of them with geographical coordinates and makes a decision to combine them by the correlated direction. With combined and generated trajectory data, this combination way allows trajectory-based navigation to guide more and better routes. In our study, this solution has been introduced. However, the ways in which correlated direction is decided and post-process works have been revised to use the sequential pattern of triangles' area GPS information between two trajectories makes in road intersection and intersection among sets comprised of GPS points. This, as a result, reduces unnecessary combinations resulting redundant outputs and enhances the accuracy of estimating correlated direction than before.

Key words: GPS Trajectory Combination, GPS Correlated Direction Estimation, Trajectory-based Navigation, Local-Based Services

1. 서 론

교통 및 지도 관련 서비스에서 도로정보는 중요하다. 대표적인 교통관련 서비스는 내비게이션으로, 도로정보를 이용하여 목적지까지 길 안내를 한다. 또 현재 지도 서비스도 단순한 위치정보제공을 넘어 경로안내에 도로정보를 사용하는 추세이다[1]. 그러나 이렇게 내비게이션에 관심도가 높아지고 경쟁이 심화되면서 이에 대한 분쟁이 생길 정도로 도로정보에 대한 개방성이 낮아지고 있다. 또한 내비게이션이 경쟁력을 위해 다양한 연구가 진행 중이다. 수집한 교통데이터의 패턴을 분석하거나 시간, 날씨 등을 고려하는 등으로 앞으로의 교통상태를 예측하는 기술이 연구되었다[1]. 수집된 경로데이터기반 내비게이션도 이러한 기술을 고려한 시스템 중 하나다. 이 내비게이션은 패턴분석을 위해 수학적 계산으로

※ Corresponding Author : Heemin Park, Address: #C303, 31 Sangmyungdae-gil, Dongnam-gu, Cheonan, TEL : +82-41-550-5367, FAX : +82-41-550-5369, E-mail : heemin@smu.ac.kr
Receipt date : Jul. 28, 2016, Approval date : Aug. 12, 2016
† Department of Computer Software Engineering, Sangmyung University (E-mail : codingple@gmail.com)
†† Department of Computer Software Engineering, Sangmyung University
※ This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (NRF-2013R1A1A1076060).

지워진 수집 경로데이터를 직접 사용하여, 실제 교통상황과 데이터를 수집한 운전자의 경험적 지식이 적용된 최적화 경로로 안내하는 것이 목적이다[2]. 수집된 경로데이터기반의 내비게이션은 한 지역에서 오랜 시간동안 운전한 운전자의 그 지역 교통상황에 대한 경험적 지식을 활용하여, 패턴분석 등의 기존의 방식이 찾을 수 없었던 최적의 경로로 안내가 가능하다. 그러나 이 시스템에서는 수집한 경로데이터를 활용하기 때문에 경로데이터의 양이 경로안내의 성능에 영향을 미친다는 점과 모든 경로에 대한 데이터 수집에는 한계가 있다는 결점이 있다. 경로조합 알고리즘은 이를 보완하기 위해 고안되었으며, 이 알고리즘을 이용하면 경로데이터끼리 조합하여 수집하지 못한 경로데이터를 얻을 수 있다. 또한 경로조합이 얻기 힘든 도로정보를 사용하지 않고 GPS 정보만으로 가능하다는 것이 특징이다.

이러한 GPS 데이터를 이용한 경로조합 알고리즘에 대한 연구는 이미 시도된 바 있다[3]. 도로정보 없이 GPS좌표의 거리에 의존하여 방향판단과 경로조합을 해야 하는 제약에도 불구하고 내비게이션이 필요로 하는 경로들을 조합할 수 있었다. 그러나 방향을 판단하는 방법과 교차로 판단방식의 한계로 중복된 경로조합과 부정확한 방향판단을 초래했다. 본 논문에서는 기존 경로조합방법을 개선하여 이 문제를 해결할 수 있는 새로운 방법을 제시한다. 경로조합 알고리즘에서 두 경로의 교차로는 각 경로의 점이 인접한 위치일 때로 가정한다. 이 때 두 점을 기준으로 두 경로의 방향에 따라 경로를 조합한다. 기존 방식에서 상호경로의 방향성 판단은 두 경로의 순차적 거리비교였다. 거리변화의 패턴이 증가와 감소의 반복패턴이면 같은 방향으로 판단하고, 증가의 반복된 패턴은 다른 방향으로 판단되었다. 여기서 거리라는 판단 기준이 다른 요소의 영향을 받으며 부정확한 판단이 발생한다. 이를 극복하기 위해 개선된 방법에서는 두 경로가 이루는 삼각형을 이용한다. 이를 통해 기존 방법에서 방향판단에 영향을 주는 요소를 제거했다. 방향판단과 조합이 끝나면 현재 교차로를 이탈하거나 다음 교차로로 가는 방법을 사용하는데, 기존에는 거리비교를 하는 각 경로의 점의 위치를 순차적으로 바꿔가며 교차로 판단이 나올 때까지 진행시켰다. 개선된 방법에서는 정해진 반경에 포함되는 점들을 공유하는 반경 안의 점들을 모두 묶었을 때, 마지막 부분에서 교차로의 거리만큼 되돌아오는 방식을 사용하여 다음 교차로까지 같은 도로로 판단하는 방법에서 기존 방법에 영향을 주는 요소를 제거했다.

2. 기존 경로조합방법

개선된 알고리즘과 기존 알고리즘은 서론에서 언급했던 교차로 식별방법과 방향성 판단방법 이외에 진행순서나 방법이 같다. 그러므로 기존 경로조합방법을 먼저 설명할 필요가 있다. 경로조합은 두 경로를 기준으로 '교차로 검출 → 상호방향성 판단 → 조합의 가치 판단 → 경로조합 → 경로조합 후 처리'의 순서다. 두 경로가 함께 통과한 교차로를 찾고, 그 교차로에서 상호방향을 판단하여 수집한 데이터와 다르고 경로안내가 가능한 새로운 경로가 조합되는지 판단하여 조합한다.

2.1 교차로 검출

교차로를 찾기 위해 한 경로를 중심으로 두 경로

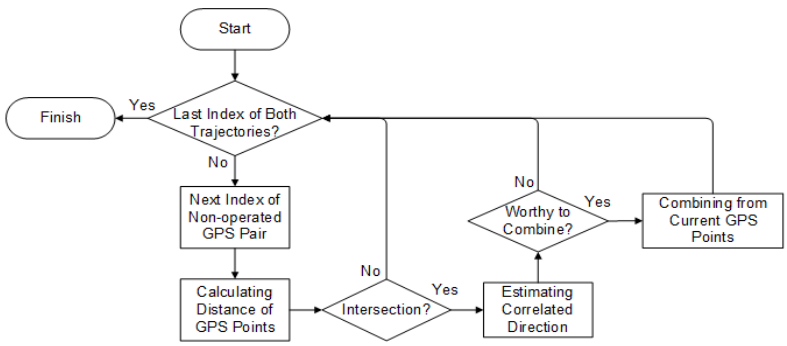


Fig. 1. Flowchart of Combining Two Trajectories [3].

의 거리를 측정한다. i번째 경로데이터 T가 포함하고 있는 n개의 GPS를 $T_i = (t_i^0, t_i^1, t_i^2 \dots t_i^{n-1})$과 같이 정의할 때, n개의 GPS를 갖는 경로데이터 T_1과 k개의 GPS를 갖는 경로데이터 T_2의 교차로 검출은 다음과 같이 나타낼 수 있다.

$$I = \left(r_1^x, r_2^y\right) \subseteq \left\{\left(T_1^x, T_2^y\right) \middle| \overline{T_1^x T_2^y} < l\right\} \quad (1)$$
$$(x = 0,1,2,\dots,n-1, y = 0,1,2,\dots,k-1)$$

I은 두 점이 같은 교차로를 이용했다고 판단할 수 있을 거리이다. I보다 작은 모든 GPS의 쌍이 I가 되지 않는 이유는 경로조합 후 처리 때문이다. 교차로로 판단된 두 점에서 경로조합이 끝나면 현재 교차로를 이탈하거나 같은 도로를 사용하여 교차로 판단을 건너뛸 때 I보다 작은 거리의 두 GPS라 해도 누락될 수 있다.

2.2 상호방향성 판단

두 점의 거리가 교차로라는 판단이 나오면 $r_1{}^x$와 $r_2{}^y$를 중심으로 두 경로의 상호방향성을 판단하게 된다. 거리측정횟수를 N으로 설정할 때, 방향성을 판단하는 방법은 다음과 같이 정의한다.

$$d_i = \overline{r_1{}^{x+\lfloor (i-1)/2 \rfloor} r_2{}^{y+\lfloor i/2 \rfloor}} \ (i = 1,2,\dots,N) \quad (2)$$

$$g(j) = d_{j+1} - d_j \ (j = 1,2,\dots,N-1) \quad (3)$$

$$D(j) = \begin{cases} increase & g(j) \geq 0 \\ decrease & g(j) < 0 \end{cases} \quad (4)$$

$D(j)$는 증가와 감소가 나열된 패턴을 나타낸다. 이 패턴이 증가와 감소가 번갈아 반복되면 두 경로는 '같은방향'으로 판단한다. 또 식 (2)를 다음과 같이 정의하고 증가-감소의 반복패턴이 나오면 '반대방향'으로 판단한다.

$$d_i = \overline{r_1{}^{x+\lfloor (i-1)/2 \rfloor} r_2{}^{y-\lfloor i/2 \rfloor}} \quad (5)$$

이 외에는 모두 '다른방향'으로 판단하나 일반적으로 증가의 연속된 패턴이 나타난다. 이렇게 두 경로의 상호방향성은 '같은방향', '반대방향', '다른방향'중에서 하나로 정의한다.

2.3 경로조합의 가치 판단

경로조합의 가치가 있다는 것은 조합한 경로가 수집한 데이터와 동일한 경로가 아니어야하고, 불필요하게 조합하는 일이 없어야하며, 경로안내가 가능하다는 것이다. 경로조합은 조합할 가치가 있을 때만 조합을 한다. 경로조합의 가치를 판단하기위한 검사항목은 다음과 같다.

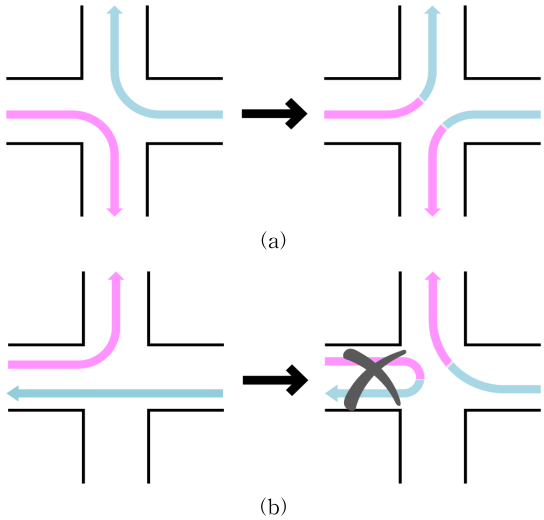


Fig. 2. Combination of (a) Both Trajectories and (b) Only One Trajectory [3].

1) T_1의 '반대방향'으로 T_2가 진행
2) T_2의 '반대방향'으로 T_1이 진행
3) T_1과 T_2가 '같은방향'으로 진행

위 항목들에서 T_1을 기준으로 교차로 검사를 진행할 때를 전제로 1)은 T_1, 2)는 T_2의 조합가치를 판단한다. 경로조합의 가치는 '두 경로 모두 조합', '한 경로만 조합', '조합하지 않음'으로 판단된다. 위 항목이 모두 거짓일 경우 서로 다른 방향으로 진행하므로 '두 경로 모두 조합', 1)이 참이면 T_2에 T_1을 조합하고 2)가 참이면 T_1에 T_2를 조합하므로 '한 경로만 조합', 3)이 참이면 두 경로 모두 해당 교차로에서 조합하지 않으므로 '조합하지 않음'이다.

2.4 경로조합 후 처리

한 교차로에서 경로조합이 끝나면 해당 교차로에서 다시 조합이 이루어질 필요가 없다. 그러나 교차로를 판단하는 기준이 거리라는 점에서 I의 크기에 따라 한 교차로에서 조합이 여러 횟수 일어날 수 있다. 이를 위해 경로조합 후에는 해당 교차로를 이탈하여 불필요한 조합을 막는다. 경로조합 후 처리를

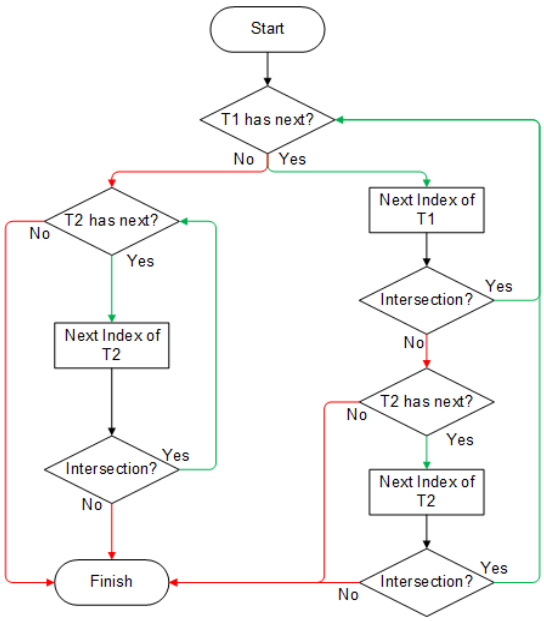


Fig. 3. Flowchart of Post-process [3].

순서도로 표현하면 다음과 같다.

또한 두 경로가 해당 교차로 이외에 다른 교차로에서 조합할 가능성이 있다면 조합의 판단에 관여하기도 하는데, 두 경로가 같은 방향으로 진행할 때와 기준 경로 T_1이 T_2의 '반대방향'으로 진행할 때이다. 예를 들어, Fig. 4는 P1에서 두 경로가 같은 교차로이지만 '같은방향'으로 판단되었으므로, 교차로를 이탈하는 처리방식으로 다음 교차로의 판단까지 연산을 줄일 수 있다. 결과적으로 T_1과 T_2는 P_2에서 두 경로 모두 조합된다.

기존 알고리즘은 경로를 조합하여 새로운 경로를 생성하였다. 그러나 교차로가 아닌 곳에서 조합이 일어나는 등 교차로 판단과 방향판단의 한계를 보였다. 이를 개선하기위하여 위 과정 중 방향을 판단하는 방법과 경로를 이탈하는 방법의 정확성을 높였다. 아래에서는 기존 알고리즘의 문제점과 이를 위한 새로운 방법을 비교하여 설명한다.

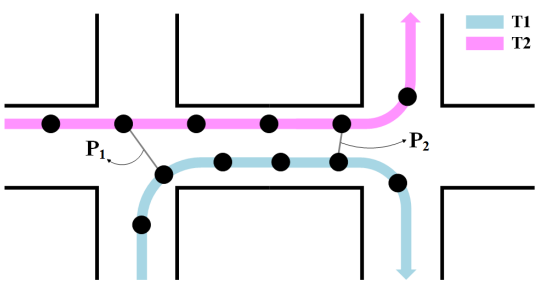


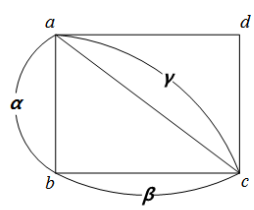
Fig. 4. Post-process in Same Direction [3].

3. 상호방향성 판단

3.1 기존 방향성판단의 문제점

기존 알고리즘에서 두 경로의 상호방향성은 GPS 순서에 따른 거리에 의존한다. $r_1{}^x$와 $r_2{}^y$를 기준으로 두 경로의 순서대로 GPS를 번갈아 옮겨가며 거리를 측정했을 때, 두 경로가 같은 방향으로 진행할 경우 증가와 감소가 반복되고 다른 방향으로 진행하면 대체적으로 증가하는 패턴을 이용했다. Fig. 5에서 $r_1{}^x$와 $r_2{}^y$가 각각 a와 b이고, $r_1{}^{x+1}$과 $r_2{}^{y+1}$이 각각 d와 c라고 가정해보자. 두 경로가 같은 방향이라면 $\overline{ad}$와 $\overline{bc}$는 평행에 가까울 것이고 $\gamma > \alpha$이므로 $\overline{ab} < \overline{ac}$로 '증가', $\overline{ac} > \overline{cd}$로 '감소'이다. 이 패턴의 반복을 감지하여 '같은방향'으로 판단하는 것이 기존 경로조합 방법의 기본적인 아이디어이다.

첫 번째 문제는 두 경로의 간격이 다를 때이다. 경로데이터는 초단위 간격으로 위치를 기록하여 수집하는데, 설정에 따라 간격의 차이가 발생할 수 있다. 또한 같은 시간간격으로 데이터를 수집해도 두 경로의 속도가 차이나면 간격이 다르다. 간격이 다를 경우 두 경로가 같은 방향으로 진행해도 거리변화패턴은 일정하지 않다. 거리변화패턴을 방향의 판단으로 사용하는 경우 이를 식별하기 힘들다. 두 번째 문제는 단순히 두 GPS 사이의 거리로 경로의 상호방향성을 판단하는 것이 불안정하다. GPS의 방향성인 두 경로의 각도를 판단해야하는 기준임에도 거리는 360° 모든 GPS들에게 공평하게 측정되어 민감하게 판단하기 힘들기 때문에 이를 보완할 수 있는



$$\gamma = \sqrt{\alpha^2 + \beta^2} > \alpha \quad (\alpha > 0, \beta > 0)$$

Fig. 5. Fundamental Idea of Direction Estimation in Prior Combination Algorithm.

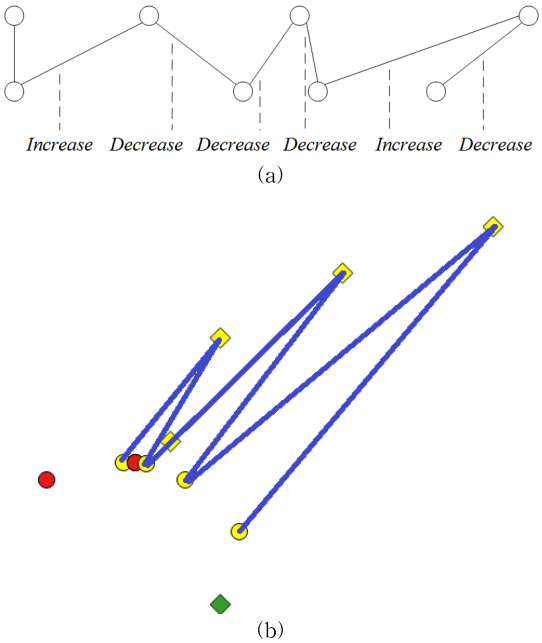


Fig. 6. Irregular GPS Interval Problem(a) and Limitation of Correlated Direction Estimation(b) in Prior Combination Algorithm.

장치를 최대한 만들어야 한다. 아래 Fig. 6의 (a)는 첫 번째 문제를 묘사한다. 두 경로가 평행하다고 가정했음에도 불구하고 GPS사이의 거리만 사용한 패턴으로는 같은 방향이라고 판단하지 못하는 경우가 생긴다. Fig. 6의 (b)는 분석한 데이터를 무료 오픈소스 지리정보시스템인 QGIS에 렌더링한 일부이다[4]. T_1과 T_2의 방향검사에서 '같은방향'과 '반대방향'이 모두 참이라는 결과가 나와 분석해보니 다른 방향임에도 '같은방향'이라는 결과가 나왔다. 차이가 근소한 이등변삼각형모양으로 왼쪽 파란색삼각형모양의 왼쪽 변부터 오른쪽 차례대로 증가-감소의 반복패턴이다.

3.2 삼각형을 이용한 상호방향성 판단

개선된 알고리즘은 먼저 기존 알고리즘과 다르게 'T_1의 반대방향으로 T_2가 진행' 항목을 검사하지 않는다. 그 이유는 T_1을 중심으로 교차로 검사가 진행되기 때문에 T_2가 T_1의 반대방향으로 진행한다면 이미 그 교차로 이전의 교차로를 검출했어야 하고 그곳에서 조합을 마쳤기 때문에 해당 결과는 나올 수 없다. 따라서 다음 두 가지 항목으로 검사한다.

1) T_1이 T_2의 정방향과 같은 방향
2) T_1이 T_2의 역반향과 같은 방향

하지만 새로운 알고리즘은 증가-감소의 패턴을 이용하여 상호방향성을 판단하는 점에서 기존 알고리즘과 흡사하다. 이 알고리즘은 $r_1{}^x$와 $r_2{}^y$부터 순서대로 삼각형을 만들어 삼각형 넓이의 변화패턴을 비교한다. 두 경로를 이루는 GPS들의 삼각형의 넓이는 서로 다른 방향일 때 지속적으로 증가하는 것을 이용했다. 또한 두 경로가 같은 방향으로 진행할 때, GPS의 간격이 일정하고 두 경로가 평행이라고 가정하면 삼각형의 넓이는 변하지 않는다. 아래 Fig. 7에서 $T_1 // T_2$, $\overline{bc} = \overline{ce} = \overline{ef}$, $\triangle abc = S1$, $\triangle cde = S2$, $\triangle efg = S3$일 때, $S1 = S2 = S3$이다.

이를 바탕으로 식 (2)를 변경하여 d_i를 다음과 같이 정의할 수 있다.

$$d_i = \triangle r_1^{x+i-1} r_1^{x+i} r_2^{y+i} \ (i = 1,2,3,\ldots,N) \tag{6}$$

식 (6)에서 삼각형의 넓이는 세 GPS의 좌표를 이용하여 구하고, T_1이 T_2의 정방향과 같은 방향인지 검사한다. 식 (3)과 식 (4)에 식 (6)에서 바뀐 d_i를 적용하여 '증가'와 '감소'의 패턴인 $D(j)$를 구한다. 여기서 $D(j)$가 일정 개수 이상의 '증가'를 가지고 있으면 T1은 T2의 정방향과 같은 방향이라고 1차 판정을 하고, 그렇지 않으면 다른 방향이라 판정한다. 그런데 삼각형의 넓이는 밑변의 길이도 영향을 미친다. 두 경로의 방향은 GPS의 간격의 영향을 받지 않아야하기 때문에 식 (6)을 다시 다음과 같이 정의할 수 있다.

$$u_i = \begin{cases} \overline{r_1^x r_1^{x+1}} & i = 1 \\ r_1^{x+i-2} r_1^{x+i-1} & i > 1 \end{cases} \tag{7}$$

$$d_i = \triangle r_1^{x+i-1} r_1^{x+i} r_2^{y+i} \times \frac{u_i}{r_1^{x+i-1} r_1^{x+i}} \tag{8}$$

$$(i = 1,2,3,\ldots,N)$$

식 (7)과 식 (8)로 $D(j)$는 삼각형의 높이변화가 된다. 1차 검사가 같은 방향으로 판정을 받으면 2차 검

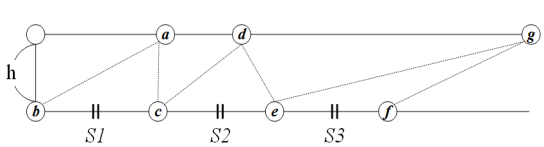


Fig. 7. Fundamental Idea of New Combination Algorithm.

사를 하는데, 삼각형의 변 중 같은 경로의 GPS를 잇는 변을 제외한 두 변의 합으로 다시 변화패턴을 확인한다. 그 이유는 정반대방향으로 진행하는 평행한 두 경로를 식별하기 위함이다. 삼각형의 크기가 변하지 않아도 두 변의 합이 연속으로 증가하는 패턴이면 다른 방향으로 판단한다. 이를 판단할 식 (2)는 다음과 같이 정의된다.

$$d_i = \overline{r_1^{x+i-1} r_2^{y+i}} + \overline{r_1^{x+i} r_2^{y+i}} \ (i = 1,2,3,...,N) \qquad (9)$$

식 (9)를 적용한 $D(j)$의 패턴도 일정 개수 이상의 '증가'를 가지면 다른 방향으로 판단하고, 2차 판정까지 다른 방향으로 판정이 나지 않으면 같은 방향으로 판단한다. 이를 순서도로 표현하면 다음과 같다.

마지막으로 상호방향성 판단에 따른 조합이다. 경로조합의 가치는 기존 알고리즘과 동일하지만 교차로에 따른 조합으로 표현만 다르게 한다. 상호방향성 판단 결과에 따라 두 경로를 r_1^x과 r_2^y에서 조합할지, 경로이탈방법을 적용한 다음 교차로에서 조합할지 결정한다. 다음 Table 1은 경로조합 판단결과를 표로 정리한 것이다.

3.3 개선된 문제

개선된 상호방향성 판단은 두 경로의 거리를 삼각형의 높이로 판단하기 때문에 기존 알고리즘에서 GPS간격이 일정하지 않을 때 생기는 문제를 해결했다. Fig. 7과 같이 두 경로는 같은 방향일 때 GPS간격의 영향을 받지 않는다. 삼각형의 밑변이 주는 영향을 없애 높이만 영향을 주므로 다른 요소의 영향을 줄이고 두 경로의 상호방향이 같을 때와 다를 때의 차이가 더 확실해졌다. 또한 Fig. 6의 (b)와 같이 GPS사이의 거리패턴으로 방향을 판단하기 어려운 부분을 해결했다. 기존 알고리즘은 방향이 다소 같으면 같은 방향으로 취급한다. 이런 면에서 개선된 상호방향성 판단은 바뀌는 두 경로의 상호방향을 더 민감하게 감지할 수 있다.

4. 경로조합 후 처리

4.1 기존 경로조합 후 처리의 문제점

교차로를 이탈하는 방법은 단순히 연산의 수를 줄이는 역할 이상으로 중요하다. 상호방향성 검사와 함께 다른 교차로를 판단하고 경로조합에 관여한다. 하지만 기존 경로조합 알고리즘에서 GPS의 위치와 거리를 이용한 경로조합 후 처리는 정확성이 낮다. 교차로를 판단하는 I은 정확성을 위해 낮아질 수밖에 없는데, I을 기준으로 다음 교차로까지 진행시키기 어렵다. 또 두 경로가 한 도로 위에 있음을 감지하기 위해서 r_1^x와 r_2^y를 기준으로 GPS의 위치를

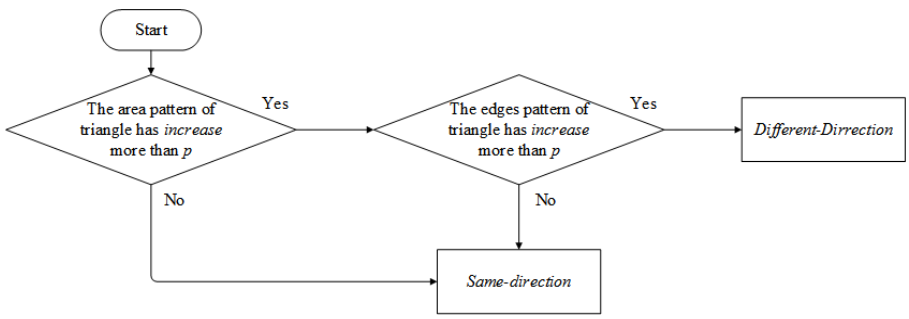


Fig. 8. Flowchart of Direction Estimation in New Algorithm.

Table 1. Correlated Direction & Combination Decision

Correlated Direction		Combined Intersection	
T_2's Front	T_2's Rear	T_1	T_2
Different	Different	This intersection	This intersection
Same	Different	Next intersection	Next intersection
Different	Same	This intersection	Next intersection
Same	Same	Algorithm error	

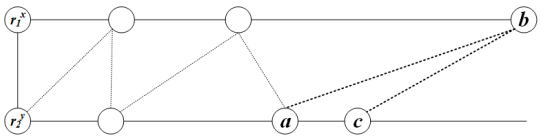


Fig. 9. Problem of Post-process in Prior Algorithm.

옮겨가며 검사를 하는데, 두 경로의 속도가 다르면 같은 도로 위의 두 경로지만 거리는 멀어질 수 있다. 이는 도로 한 가운데를 교차로라고 판단하는 문제를 야기한다.

4.2 개선된 경로조합 후 처리

먼저 교차로를 판단하는 l만 이용하지 않고, l보다 포용력 있는 범위로 같은 도로임을 검사할 수 있는 거리인 L을 따로 설정한다. 경로조합이 끝나고 $r_1{}^x$를 중심으로 반경 L안에 있는 T_2의 GPS들로 집합을 만든다. 다음 GPS인 $r_1{}^{x+1}$에서도 같은 방법으로 T_2의 GPS집합을 만든다. 이 때 두 집합의 교집합이 공집합이 아니면 두 집합을 합치고 $r_1{}^{x+2}$에서도 같은 방법으로 교집합을 검사하고 합집합으로 만든다.

이렇게 교집합이 공집합이 될 때까지 만들어진 집합 안에서 T_1의 가장 큰 인덱스부터 거리가 l보다 작은 T_2의 GPS를 찾아 작은 인덱스로 옮겨간다. 이 때 거리를 이용하여 집합을 만들었기 때문에 다른 고도의 도로 등 다른 도로의 GPS도 묶일 수 있으므로 집합 안에 연속된 GPS로 다시 집합을 나누고, 인덱스를 옮기며 l보다 짧은 거리의 GPS쌍을 찾는 검사는 $r_1{}^y$를 포함한 집합 안에서 이루어진다. 이 두 GPS는 $r_1{}^x$와 $r_2{}^y$에서 조합이 끝나고 교차로 검사를 시작해야 할 위치이거나 다음 교차로가 된다. 이 두 점을 $r_1{}^{last}$와 $r_2{}^{last}$라 하고, 교집합이 공집합이 나올 때까지 검사횟수를 q라 할 때 다음과 같이 정의할 수 있다.

$$C_i = \left\{ r_1^{x+i}, r_2^a \mid \overline{r_1^{x+i} r_2^a} < L \right\} (i = 0,1,2,...,q \quad \text{그리고} a = 0,1,2,...,k-1) \tag{10}$$

$$C_i \cap C_{i+1} \neq \phi (i = 0,1,2,...,q-1) \tag{11}$$

$$W = C_0 \cup C_1 \cup C_2 \cup ... \cup C_q \tag{12}$$

$$r_1^{last}, r_2^{last} \in W \tag{13}$$

$r_1{}^{last}$와 $r_2{}^{last}$를 구하는 이유는 L이 l보다 범위를 넓히기 위해 만든 기준이기 때문에, W의 두 경로의 마지막 인덱스를 조합하는 것은 부자연스러운 경로가 될 수 있다. 따라서 최대한 가까운 거리로 좁힌 두 GPS에서 진행하는 것이다.

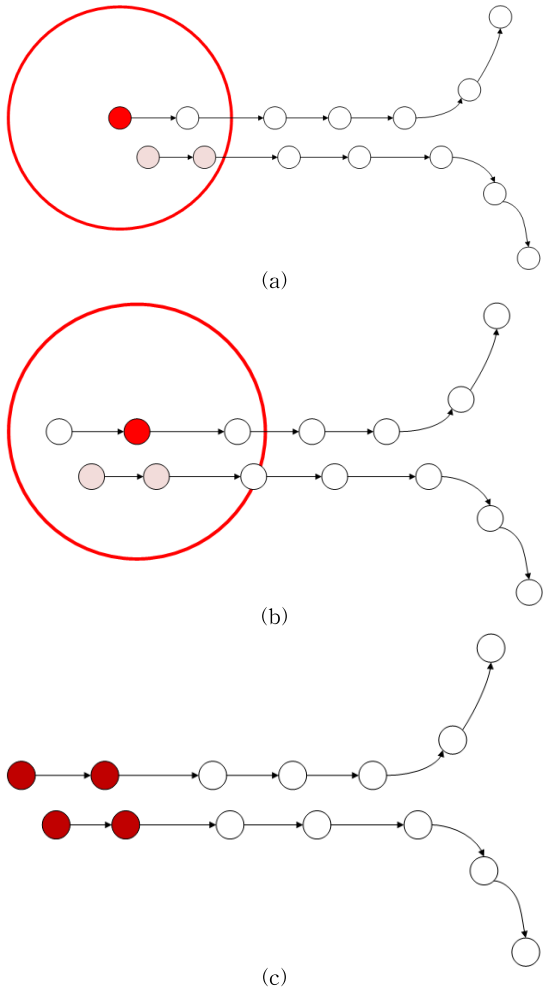


Fig. 10. Partial Step of New Post-process. First Set (a), Next Set (b), and Merged Set (c).

4.3 개선된 문제

기존 알고리즘에서 가장 성능을 향상시킨 부분이 경로조합 후 처리방법이다. 고속도로와 같이 속도가 빠른 도로에서 GPS사이의 간격은 더 벌어지기 때문에 l로 판단하는 데에 한계가 크다. 실제 기존 알고리즘의 결과를 분석했을 때, 원하는 경로의 조합은 이루어졌지만 그 교차로를 벗어나지 못하고 조합이 반복되어 불필요한 경로생성과 연산을 초래했다. 특

히 넓은 범위를 고려한 거리를 이용했음에도 목표한 위치의 도로를 찾아내어 비교하는 방법으로 정확성을 올릴 수 있었다.

5. 실험 결과 및 결론

5.1 실험 환경

먼저 실험을 위한 경로데이터는 직접 수집한 데이터로, Table 2와 같이 기존 알고리즘의 실험에서 사용한 데이터이다. 경로데이터는 GPS정보의 XML 형식인 GPX[5]이며 참여형 오픈소스 지도서비스인 오픈스트리트맵[6]을 기반으로 한 개발된 OsmAnd [7]을 사용했다. Fig. 11은 수집한 경로데이터를 오픈스트리트맵 기반 맵에디터인 uMap[8]으로 지도 위에 표현했으며, 경로가 겹칠수록 색이 진해진다.

Table 2. Collected Trajectory Data [3]

Collected Trajectory Data Information	
Number of Trajectories	56
File Size	13.9 MB
Area	from Seoul to Jeonju (Korea)
Used Navigation Systems	T-map, Kimgisa, iphone

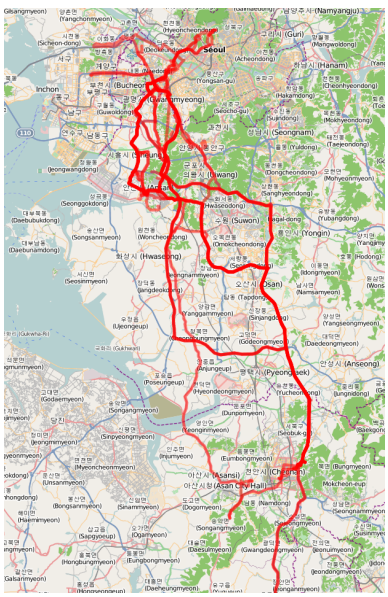


Fig. 11. Illustration of Collected Trajectories with uMap [3].

알고리즘은 스크립트 언어인 Python[9]으로 구현하였다. 또한 Window OS에서 실험하였으며, 실험 환경에 대한 자세한 내용은 Table 3와 같다.

Table 3. Experimental Environments

Specification of Experimental Machine	
Main Memory	4 GB
Processor	Intel(R) Core(TM) i3-4005U 1.70GHz
OS	Windows 10 Home (64bit)
Python Version	2.7.11

5.2 실험 결과

실험에서 사용된 기준계수는 I=10m, L=100m, N=5이며, 패턴에서 '증가'가 75%, 즉 3개 이상일 경우 '다른방향'으로 판단했다. 실험은 안전한 실험을 위해 교차로 검사의 결과를 MongoDB[10]에 먼저 저장한 후, 방향성 검사와 경로조합 순서로 진행했다. 교차로 검사의 결과로 데이터베이스에 저장되는 내용은 상호방향성 검사를 가능하게 할 데이터들로 조합할 두 경로의 이름, (r_1^x, r_2^y)와 방향성 검사를 할 그 주변의 N개의 GPS, 그리고 (r_1^{last}, r_2^{last})이다. 교차로 검사의 결과는 다음과 같다.

Table 4. Results of Intersection Estimation

Intersection Experiment	
Number of Records	1,797
Run Time	1h 51m

데이터베이스에 저장된 정보로 상호방향성 판단 및 조합 결과는 다음과 같다.

Table 5. Results of Direction Estimation

T_2's Front	T_2's Rear	Number of Output
Different	Different	1639
Same	Different	107
Different	Same	51
Same	Same	0

5.2 결론

다음 Fig. 12는 특정한 하나의 경로를 중심으로

Table 6. Results of Trajectory Combination

Output Information	
Number of Total Trajectories	3,594
File Size	1.36 GB

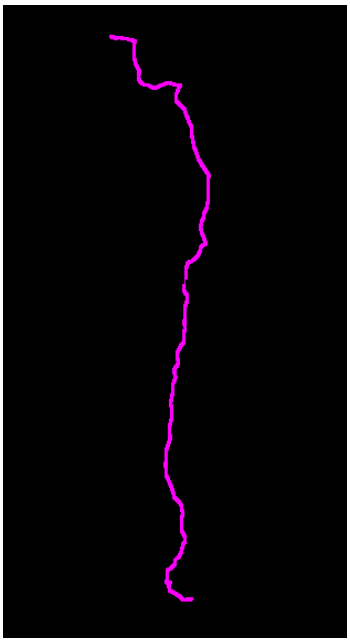


Fig. 12. Combined Result of Specified Trajectory.

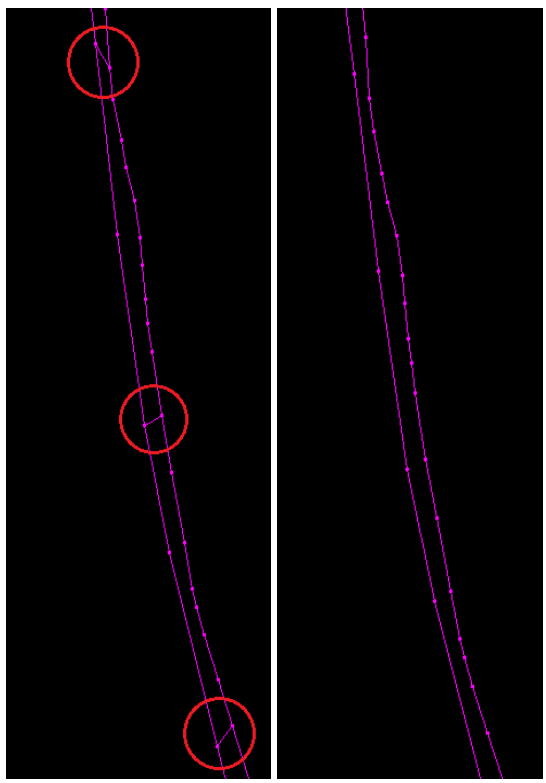
Table 7. Number of Combined Result with Specified Trajectory

Output Information		
	Prior Algorithm	New Algorithm
Number of Total Output	19,168	3,594
File Size of Total Output	7.11 GB	1.36 GB
Number of Combined Trajectories with Above Route	186	17

조합된 경로의 결과를 오픈스트리트맵기반 맵에디터인 JOSM[11]에서 가시화한 것이다. 멀리서 봤을 때 기존 알고리즘과 새로운 알고리즘이 똑같은 모양으로 그려진다. 그러나 결과적으로 원하는 경로를 똑같이 얻었음에도 생성된 경로는 17개로, 이전 알고리즘의 약 9.1%에 불과했다. 이렇듯 전체결과에서 조합된 경로의 개수는 18.75%로 줄었으며 용량은 약 19.13%로 줄었다.

이를 자세히 보기 위해, 다음 Fig. 13은 Fig. 12의 일부를 확대한 그림으로 (a)는 기존 알고리즘의 결과이고 (b)는 새로운 알고리즘의 결과를 가시화하였다. (a)에서는 같은 도로임에도 두 경로의 GPS간격 차로 인한 알고리즘의 결함으로 부정확한 경로조합이 생긴 흔적이 보이지만, (b)에서는 보이지 않는다. 따라서 새로운 알고리즘이 더 정확한 판단을 하고 있음을 알 수 있다.

위의 결과가 보여주듯이, 알고리즘의 성능 개선으로 원하는 조합경로는 그대로 얻으며 기존에 부정확한 조합이 만들어낸 경로들을 제거하여 조합경로의 개수와 용량을 기존의 20% 이하로 줄일 수 있었다. 또 기존 알고리즘에서는 Fig. 6의 (b)와 같이 예상하지 못한 오류의 결과를 보였으나 개선된 알고리즘에서는 Table 5와 같이 불가능한 결과가 나오지 않았다. 그러나 새로운 알고리즘에서도 판단이 '패턴의 증가가 75% 이상'에 따라 결정되는 것과 같이 유연하지 못한 기준을 유연하게 만드는 등 더 개선하여 정확성을 높일 수 있을 것으로 보인다. 따라서



(a) (b)

Fig. 13. Detail Result of Fig. 12.

앞으로는 본 논문에서 제안한 알고리즘보다 정확도가 높은 알고리즘을 고안하고, 실제 내비게이션 시스템에 적용을 해볼 예정이다.

REFERENCE

[1] Advanced Traffic Prediction of T-map, http://readme.skplanet.com/?p=8870, (accessed July, 25, 2016).

[2] K.M. Koo, T.H. Lee, and H.M. Park, "Navigation System using Big Trajectory Data based on Empirical Knowledge," *Proceeding of the Summer Conference of Korea Information Science Society*, pp. 2150-2152, 2015.

[3] K.M. Koo, T.H. Lee, and H.M. Park, "A Big-Data Trajectory Combination Method for Navigations using Collected Trajectory Data," *Journal of Korea Multimedia Society*, Vol. 19, No. 2, pp. 386-395, 2016.

[4] QGIS, https://www.qgis.org, (accessed July, 25, 2016).

[5] GPX: The GPX Exchange Format, http://www.topografix.com/gpx.asp, (accessed July, 25, 2016).

[6] OpenStreetMap, https://www.openstreetmap.org, (accessed July, 25, 2016).

[7] OsmAnd: Offline Mobile Maps & Navigation, http://osmand.net/, (accessed July, 25, 2016).

[8] uMap, http://umap.openstreetmap.fr/en/, (accessed July, 25, 2016).

[9] Python, https://www.python.org/, (accessed July, 25, 2016).

[10] MongoDB, https://www.mongodb.com/, (accessed July, 25, 2016).

[11] JOSM, https://josm.openstreetmap.de/, (accessed July, 25, 2016).

구 광 민

2015년 상명대학교 컴퓨터소프트웨어공학과 (학사)

관심분야 : 빅데이터 처리, 인공지능, 기계학습

박 희 민

1993년 서강대학교 전자계산학과 (학사)

1995년 서강대학교 컴퓨터공학과 (석사)

2006년 UCLA 전자공학과 (박사)

2013년-현재 상명대학교 컴퓨터소프트웨어공학과 조교수

관심분야 : 웹기반정보시스템, 빅데이터처리, 사물인터넷